

Scalable stereo audio encoding/decoding method and apparatus

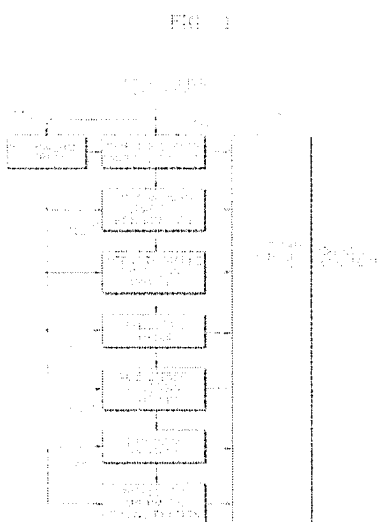
Patent number:	CN1218334 (A)	Also published as:	
Publication date:	1999-06-02		CN1126265 (C)
Inventor(s):	PARK SUNG-HEE [KR]; KIM YEON-BAE [KR]		EP0918407 (A2)
Applicant(s):	SAMSUNG ELECTRONICS CO LTD [KR]		EP0918407 (A3)
Classification:			EP0918407 (B1)
- international:	G11B20/10; G10L19/00; H03H7/38; H03M7/30; H03M7/38; H03M7/40; H04B1/66; H04H20/47; G11B20/10; G10L19/00; H03H7/38; H03M7/30; H03M7/38; H03M7/40; H04B1/66; H04H1/00; H04H5/00; (IPC1-7): H03M7/30		US6529604 (B1)
- european:	H04H20/88; H04B1/66		RU2197776 (C2)
Application number:	CN19981015468 19980709		KR100335611 (B1)
Priority number(s):	KR19970061605 19971120		JP11317672 (A)
			JP11317672 (A)
			JP3412082 (B2)

<< less

Abstract not available for CN 1218334 (A)

Abstract of correspondent: **EP 0918407 (A2)**

A scalable stereo audio encoding/decoding method and apparatus are provided. The method includes the steps of signal-processing input audio signals and quantizing the same for each predetermined coding band, coding the quantized data corresponding to the base layer among the quantized data, coding the quantized data corresponding to the next enhancement layer of the coded base layer and the remaining quantized data uncoded due to a layer size limit and belonging to the coded layer, and sequentially performing the layer coding steps for all enhancement layers to form bitstreams, wherein the base layer coding step, the enhancement layer coding step and the sequential coding step are performed such that the side information and quantized data corresponding to a layer to be coded are represented by digits of a same predetermined number; ; and then arithmetic-coded using a predetermined probability model in the order ranging from the MSB sequences to the LSB sequences, bit-sliced left-channel data and right-channel data being alternately coded in units of predetermined vectors.



Data supplied from the esp@cenet database — Worldwide

[19]中华人民共和国国家知识产权局

[51]Int. Cl⁶

H03M 7/30

[12] 发明专利申请公开说明书

[21] 申请号 98115468.9

[43]公开日 1999年6月2日

[11]公开号 CN 1218334A

[22]申请日 98.7.9 [21]申请号 98115468.9

[30]优先权

[32]97.11.20 [33]KR [31]61605/97

[71]申请人 三星电子株式会社

地址 韩国京畿道

[72]发明人 朴成熙 金延培

[74]专利代理机构 中国国际贸易促进委员会专利商标事务所

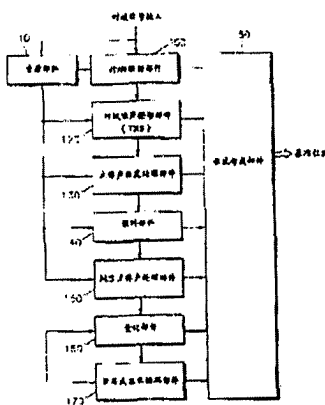
代理人 郑 迅

权利要求书 6 页 说明书 52 页 附图页数 4 页

[54]发明名称 可伸缩的立体声音频编码/解码方法和装置

[57]摘要

提供一种可伸缩的立体声编码/解码方法和装置。该方法包括以下步骤:对各预定编码频带的音频输入信号进行信号处理和量化,对量化数据中与基准层相对应的量化数据进行编码,对对应于基准层的下一增强层的量化数据进行编码,和对由于层数限制未编码而又属于编码层的剩余量化数据进行编码,对所有增强层顺序执行层编码步骤以便形成位流;随后按照从 MSB 序列到 LSB 序列的次序,使用先验概率模型进行算术编码,以预定向量组轮流对经过位片处理的左声道数据和右声道数据进行解码。



ISSN 1008-4274

权 利 要 求 书

1.一种可伸缩的立体声音频编码方法,该方法将音频信号编码为具有一层基准层和至少两层增强层的分层数据流,该方法包括以下步骤:

对各预定编码频带的音频输入信号进行信号处理和量化;

对量化数据中与基准层相对应的量化数据进行编码;

对对应于基准层的下一增强层相对应的量化数据进行编码,以及对由于层数限制未编码而又属于编码层的剩余量化数据进行编码;以及

对所有增强层顺序执行层编码步骤以便形成位流,其中执行基准层编码步骤、增强层编码步骤以及顺序编码步骤,从而可以利用相同的预定数字表示边信息以及与编码层相对应的量化数据;随后按照从 MSB 序列到 LSB 序列的次序,利用先验概率模型进行算术编码,以预定向量组轮流对位片式左声道数据和右声道数据进行编码。

2.根据权利要求1的可伸缩的立体声音频编码方法,其中边信息至少包括比例因子和算术编码所使用的各概率模型的信息。

3.根据权利要求1的可伸缩的立体声音频编码方法,其中预定向量是通过将四个位片式音频声道数据耦合为一个向量而生成的四维向量。

4.根据权利要求3的可伸缩的立体声音频编码方法,其中根据预状态将四维向量划分为两个子向量,以便进行编码,其中预状态表示是否对非零位片式频率分量进行过编码。

5.根据权利要求2的可伸缩的立体声音频编码方法,其中对比例因子进行编码的步骤包括以下步骤:

获得最大比例因子;

获得最大比例因子与第一比例因子的差值,对该差值进行算术编码;

以及

获得紧接前一个算术编码的比例因子与第一比例因子之后各比例因子的差值,将该差值映射为预定值,并对映射值进行算术编码。

6.根据权利要求5的可伸缩的立体声音频编码方法,其中在映射步骤中使用表 5.1 所列的概率模型。

7.根据权利要求5的可伸缩的立体声音频编码方法,其中在算术编码步

骤中使用表 5.3 到表 5.4 所列的概率模型。

8.根据权利要求 2 的可伸缩的立体声音频编码方法,其中对比例因子进行编码的步骤包括以下步骤:

获得最大比例因子; 以及

获得最大比例因子与各比例因子的差值, 并对该差值进行算术编码。

9.根据权利要求 1 的可伸缩的立体声音频编码方法,其中为了得到分层结构,对通常用于所有频带的标题信息进行编码,随后对利用位片信息形成的各层所需的边信息和量化频率进行编码。

10.根据权利要求 1 的可伸缩的立体声音频编码方法,其中通过以下步骤完成量化:

将时域上的音频输入信号转换为频域信号;

利用时间/频率映射以预定比例因子频带信号的方式耦合转换信号,计算各比例因子频带的屏蔽阈值;

进行时域噪声整形,以便控制各转换窗口内量化噪声的瞬时形状;

进行立体声强度处理,从而仅对两条声道中一条声道的比例因子频带的量化信息进行编码,并且仅传输另一条声道的比例因子;

预测当前帧的频率系数;

进行 M/S 立体声处理,以便将左声道信号和右声道信号转换为两个信号的相加信号以及其相减信号; 和

对各预定编码频带的信号进行量化,从而各频带的量化噪声小于屏蔽阈值。

11.根据权利要求 1 的可伸缩的立体声音频编码方法,其中当量化数据是由符号数据和数值数据组成时,基准层和增强层编码以及位流形成步骤还包括以下步骤:

对由数值数据的最高有效数字构成的最高有效数字序列进行算术编码;

对与编码最高有效数字序列中非零数据相对应的符号数据进行编码;

对数字数据的未编码数值数据中的最高有效数值序列进行编码;

对编码与编码数字序列中非零数值数据相对应的符号数据中的未编码符号数据进行编码; 以及

对数字数据的各数字执行数值编码步骤和符号编码步骤，按预定向量组对左声道数据和右声道数据轮流执行各步骤。

12.一种可伸缩的音频编码装置，该装置包括：

量化部件，该部件用于各编码频带的音频输入信号的信号处理和量化；

位片式算术编码部件，该部件用于编码所有层的位流从而具有分层结构，通过限制基准层的频带实现可伸缩，该部件还用于编码与基准层相对应的边信息，从最高有效位序列到最低有效位序列、从较低频率分量到较高频率分量对量化信息进行顺序编码，以预定向量组轮流对左声道数据和右声道数据进行编码，对与基准层的下一增强层相对应的边信息和量化数据进行编码；以及

位流形成部件，该部件用于收集量化部件和位片式算术编码部件生成的数据并且生成位流。

13.根据权利要求12的可伸缩的音频编码装置，其中量化部件包括：

时间/频率映射部件，该部件将时域上的音频输入信号转换位频域信号；

音质部件，该部件通过时间/频率映射利用预定比例因子频带的信号连接转换信号，该部件还利用各信号相互影响而产生的屏蔽现象计算各比例因子频带的屏蔽阈值；以及

量化部件，当比较各频带的量化噪声与屏蔽阈值时，该部件量化各预定编码频带的信号。

14.根据权利要求13的可伸缩的音频编码装置，该装置还包括：

时域噪声整形（TNS）部件，该部件进行时域噪声整形以便控制各转换窗口内量化噪声的瞬时形状；

立体声强度处理部件，该部件进行立体声强度处理，从而仅对两个声道中之一个声道的比例因子频带的量化信息进行编码，传输另一个声道的比例因子；

预测部件，该部件预测当前帧的频率系数；以及

M/S 立体声处理部件，该部件进行 M/S 立体声处理，以便将左声道信号和右声道信号转换为两个信号的相加信号及其相减信号。

15.一种可伸缩的立体声音频解码方法，该方法用于解码音频数据编码以便具有分层位率，该方法包括以下步骤：

分析具有分层结构的位流中各模块所需的数据；

按具有分层结构的位流中各层的建立次序，至少解码比例因子、算术编码模型系数和量化数据，其中按照从较高有效位到较低有效位的顺序，通过分析构成位流的位的有效性，轮流对各声道的量化数据进行解码；

将经过解码的比例因子和量化数据恢复位具有原始数值的信号；以及将量化信号反演为时域信号。

16.根据权利要求15的可伸缩的立体声音频解码方法，该方法还包括以下步骤：

进行M/S立体声处理，以便检查是否在位流编码方法中进行过M/S立体声处理，并且如果进行过M/S立体声处理，就将左声道信号和右声道信号转换为两个信号的相加信号以及其相减信号；

检测是否在位流编码方法中执行过预测步骤，并且如果执行过预测步骤，就预测当前帧的频率系数；

检测是否在位流编码方法中进行过立体声强度处理，并且如果进行过立体声强度处理，则由于仅对两条声道中之一条声道（左声道）的比例系数频带的量化信息进行过编码，所以进行立体声强度处理以便将另一声道（右声道）的量化信息恢复到左声道值；以及

检测是否在位流编码方法中执行过时域噪声整形（TNS）步骤，并且如果执行过TNS步骤，就进行时域噪声整形以便控制各转换窗口内量化噪声的瞬时形状。

17.根据权利要求15或权利要求16的可伸缩的立体声音频解码方法，其中当量化数据是由符号数据和数值数据组成时，通过顺序解码量化频率分量符号位的数值数据并且耦合数值数据和符号位，恢复量化频率分量。

18.根据权利要求15的可伸缩的立体声音频解码方法，其中按照从最高有效位到最低有效位的顺序执行解码步骤，通过耦合经过解码的位片式数据以及将耦合数据恢复为量化频率分量数据，执行恢复步骤。

19.根据权利要求18的可伸缩的立体声音频解码方法，其中在解码步骤中对数据进行解码，从而将四个样本的位片式信息解码为四维向量组。

20.根据权利要求 19 的可伸缩的立体声音频解码方法,其中进行四维向量解码,从而对根据预状态编码的两个子向量进行算术解码,并且将根据各样本的编码状态进行解码的两个子向量恢复为四维向量,其中预状态表示是否对非零位片式频率分量进行过编码。

21.根据权利要求 17 的可伸缩的立体声音频解码方法,其中当根据 MSB 对各频率分量是位片式数据进行解码时,如果位片式数据为“0”就跳过解码,当位片式数据“1”第一次出现时,就对符号数据进行算术解码。

22.根据权利要求 15 的可伸缩的立体声音频解码方法,其中通过解码位流中的最大比例因子,算术解码最大比例因子和各比例因子的差值,从最大比例因子中减去该差值,就完成了比例因子的解码。

23.根据权利要求 15 的可伸缩的立体声音频解码方法,其中解码比例因子的步骤包括以下步骤:

根据位流解码最大比例因子;

通过映射获得最大比例因子与需要解码的比例因子之间的差值,算术解码该差值,根据映射值,对该差值进行逆映射;以及

通过从最大比例因子中减去该差值获得第一比例因子,而通过从前一比例因子中减去该差值获得剩余频带的比例因子。

24.根据权利要求 15 的可伸缩的立体声音频解码方法,其中通过执行以下步骤完成算术编码模型系数的解码:

解码位流中的最小算术模型系数,解码最小系数与各层边信息中各系数之间的差值,累加最小系数与该差值。

25.一种可伸缩的音频解码装置,该装置用于解码音频数据编码以便具有分层位率,该装置包括:

位流分析部件,该部件分析具有分层结构的位流中各模块所需的数据;

解码部件,该部件按具有分层结构的位流中各层的建立次序,至少解码比例因子、算术编码模型系数和量化数据,其中按照从较高有效位到较低有效位的顺序,通过分析构成位流的位的有效性,轮流对各声道的量化数据进行解码;

恢复部件，该部件将经过解码的比例因子和量化数据恢复位具有原始数值的信号；以及

频率/时间映射部件，该部件将量化信号反演为时域信号。

26.根据权利要求 25 的可伸缩的立体声音频解码装置，该装置还包括：

进行 M/S 立体声处理的 M/S 立体声处理部件，以便检查是否在位流编码方法中进行过 M/S 立体声处理，并且如果进行过 M/S 立体声处理，就将左声道信号和右声道信号转换为两个信号的相加信号以及其相减信号；

预测部件，该部件用于检测是否在位流编码方法中执行过预测步骤，并且如果执行过预测步骤，就预测当前帧的频率系数；

立体声强度处理部件，该部件用于检测是否在位流编码方法中进行过立体声强度处理，并且如果进行过立体声强度处理，则由于仅对两条声道之一条声道（左声道）的比例系数频带的量化信息进行过编码，所以进行立体声强度处理以便将另一声道（右声道）的量化信息恢复到左声道值；以及

时域噪声整形（TNS）部件，该部件用于检测是否在位流编码方法中执行过时域噪声整形步骤，并且如果执行过 TNS 步骤，就进行瞬时噪声处理以便控制各转换窗口内量化噪声的瞬时形状。

说明书

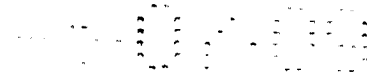
可伸缩的立体声音频编码/解码 方法和装置

本发明涉及音频编码和解码方法，更确切地说，涉及采用位片式算术编码方法的可伸缩的立体声音频编码/解码方法和装置。

在常规可伸缩音频编码/解码装置中，人们考虑了 1-声道单声道信号的可伸缩性[K. Brandenbrug 等，“可伸缩音频编码的主要思想”，97th AES-Convention, preprint 3294, San Francisco, 1994]和[K. Brandenbrug 等，“2 级或 3 级位率的可伸缩音频编码系统”，99th AES-Convention, preprint 4132, New York, 1995]。然而，MPEG 音频标准[MPEG 委员会 ISO/IEC/JTC1/SC29/WG11，信息技术-以大约 1.5 Mbit/s 速率对用于数据存储介质的动画以及各音频信号进行编码 - 第三部分：音频，ISO/IEC IS 11172-3，1998]或 AC-2/AC-3 方法[Dolby，“Dolby AC-3 多声道音频编码方法-提交给 Grand Alliance Specialist Group”，Dolby 实验室，1993 年 8 月]提供了处理立体声和多声道信号以及单声道信号的技术。实际上，大部分音乐信号是由立体声信号组成的。因此，如同因特网或通信网一样，有必要采用两声道以上位流信号沿用的可伸缩的音频编码解码器。

通常，音乐信号为立体声信号。通过光盘（CD）、通信网络或广播网络提供立体声信号，而且未来仍将在多媒体环境下提供立体声信号。然而，现有可伸缩的音频编码解码器主要处理单声道信号，而并不处理立体声信号。为了处理立体声信号，必须按以下方式传输信号，即传输某声道的所有信号，然后传输另一声道的所有信号。然而，对于以上情况，由于两个声道中产生的位数通常并不相等，所以当立体声信号的位率较低时，可伸缩音频编码解码器的性能非常低。

为了解决以上问题，本发明的目的在于提供一种可伸缩的立体声数字音频数据编码方法和装置，以及记录该编码方法的记录介质。利用位片式算术编码（BSAC）技术，通过生成由基于基准层的若干增强层组成的位流进行编码。



为了实现本发明的目的，本文提供可伸缩的立体声音频编码方法，该方法将音频信号编码为具有一层基准层和至少两层增强层的分层数据流，该方法包括以下步骤：对各预定编码频带的音频输入信号进行信号处理和量化，对量化数据中与基准层相对应的量化数据进行编码，对对应于基准层的下一增强层的量化数据进行编码，以及对由于层数限制未编码而又属于编码层的剩余量化数据进行编码，并且顺序对所有增强层顺序执行层编码步骤以便形成位流，其中执行基准层编码步骤、增强层编码步骤以及顺序编码步骤，从而可以利用相同的预定数字表示边信息以及与编码层相对应的量化数据，随后按照从最高有效位（MSB）序列到最低有效位（LSB）序列的次序，使用先验概率模型进行算术编码，以预定向量组轮流对位片式声道数据和右声道数据进行编码。边信息至少包括比例因子和算术编码所使用的各概率模型的信息。预定向量是通过将四个位片式音频声道数据耦合为一个向量而生成的四维向量，根据预状态将四维向量划分为两个子向量，以便进行编码，其中预状态表示非零位片式频率分量是否进行过编码。

此外，对比例因子进行编码的步骤包括以下步骤：获得最大比例因子，获得最大比例因子与第一比例因子的差值，对该差值进行算术编码，获得紧接前一个算术编码的比例因子与第一比例因子之后各比例因子的差值，将该差值映射为预定值，并对映射值进行算术编码。

对比例因子进行编码的步骤包括以下步骤：获得最大比例因子，获得最大比例因子与各比例因子的差值，以及对该差值进行编码。

为了得到分层结构，对通常用于所有频带的标题信息进行编码，随后对利用位片信息形成的各层所需的边信息和量化频率进行编码。

通过以下步骤完成量化：将时域上的音频输入信号转换为频域信号，经过时间/频率映射以预定比例因子频带信号的方式连接转换信号，计算各比例因子频带的屏蔽阈值，进行时域噪声整形以便控制各转换窗口内量化噪声的瞬时形状，进行立体声强度处理，从而仅对两条声道中一条声道的比例因子频带的量化信息进行编码，并且仅传输另一条声道的比例因子，预测当前帧的频率系数，进行Mid/Side（M/S）立体声处理，以便将左声道信号和右声道信号转换为两个信号的相加信号以及其相减信号，对各预

定编码频带的信号进行量化，从而各频带的量化噪声小于屏蔽阈值。

当量化数据是由符号数据和数值数据组成时，基准层和增强层编码以及位流形成步骤还包括以下步骤：对由数值数据的最高有效数字构成的最高有效数字序列进行算术编码，在最高有效数字序列中的编码对应非零数据的符号数据进行算术编码，在数字数据的未编码数值数据中对最高有效数字序列进行编码，在已编码数字序列中对对应于非零值数据一的符号数据中的未编码符号数据进行编码，对数字数据的各数字执行数值编码步骤和符号编码步骤，按预定向量组对左声道数据和右声道数据轮流执行各步骤。

可伸缩的立体声音频解码装置还包括：进行 M/S 立体声处理的 M/S 立体声处理部件，以便检查是否在位流编码方法中进行过 M/S 立体声处理，并且如果进行过 M/S 立体声处理，就将左声道信号和右声道信号转换为两个信号的相加信号及其相减信号，预测部件，该部件用于检测是否在位流编码方法中执行过预测步骤，并且如果执行过预测步骤，就预测当前帧的频率系数，立体声强度处理部件，该部件用于检测是否在位流编码方法中进行过立体声强度处理，并且如果进行过立体声强度处理，则由于仅对两条声道之一声道（左声道）的比例系数频带的量化信息进行过编码，所以进行立体声强度处理以便将另一声道（右声道）的量化信息恢复到左声道值，和时域噪声整形（TNS）部件，该部件用于检测是否在位流编码方法中执行过时域噪声整形步骤，并且如果执行过 TNS 步骤，就进行瞬时噪声处理以便控制各转换窗口内量化噪声的瞬时形状。

根据本发明的另一方面，提供一种可伸缩的音频编码装置，该装置包括：量化部件，该部件用于各编码频带的音频输入信号的信号处理和量化，位片式算术编码部件，该部件用于编码所有层的位流从而具有分层结构，通过限制基准层的频带实现可伸缩，该部件还用于对对应于基准层的边信息进行编码，从最高有效位序列到最低有效位序列、从较低频率分量到较高频率分量对量化信息进行顺序编码，以预定向量组轮流对左声道数据和右声道数据进行编码，并且对对应于基准层的下一增强层一的边信息和量化数据进行编码，以及位流形成部件，该部件用于收集量化部件和位片式算术编码部件生成的数据并且生成位流。

量化部件包括时间/频率映射部件，该部件将时域上的音频输入信号转换为频域信号，音质部件，该部件通过时间/频率映射利用预定比例因子频带的信号连接转换信号，该部件还利用各信号相互影响而产生的屏蔽现象计算各比例因子频带的屏蔽阈值，以及量化部件，当比较各频带的量化噪声与屏蔽阈值时，该部件量化各预定编码频带的信号。

此外，该装置还包括时域噪声整形（TNS）部件，该部件进行时域噪声整形以便控制各转换窗口内量化噪声的瞬时形状，立体声强度处理部件，该部件进行立体声强度处理，从而仅对两个声道中之一个声道的比例因子频带的量化信息进行编码，传输另一个声道的比例因子，预测部件，该部件预测当前帧的频率系数，以及 M/S 立体声处理部件，该部件进行 M/S 立体声处理，以便将左声道信号和右声道信号转换为两个信号的相加信号及其相减信号。

根据本发明的另一方面，提供一种可伸缩的立体声音频解码方法，该方法用于解码音频数据编码以便具有分层位率，该方法包括以下步骤：分析具有分层结构的位流中各模块所需的数据，以具有分层结构的位流中各层的建立次序，至少对比例因子、算术编码模型系数和量化数据进行解码，其中按照从较高有效位到较低有效位的顺序，通过分析构成位流的位的有效性，轮流对各声道的量化数据进行解码，将经过解码的比例因子和量化数据恢复位具有原始数值的信号，以及将量化信号反演为时域信号。

可伸缩的立体声音频解码方法还包括以下步骤：进行 M/S 立体声处理，以便检查是否在位流编码方法中进行过 M/S 立体声处理，并且如果进行过 M/S 立体声处理，就将左声道信号和右声道信号转换为两个信号的相加信号及其相减信号，检测是否在位流编码方法中执行过预测步骤，并且如果执行过预测步骤，就预测当前帧的频率系数，检测是否在位流编码方法中进行过立体声强度处理，并且如果进行过立体声强度处理，则由于仅对两条声道中之一条声道（左声道）的比例系数频带的量化信息进行过编码，所以进行立体声强度处理以便将另一声道（右声道）的量化信息恢复到左声道值，检测是否在位流编码方法中执行过时域噪声整形（TNS）步骤，并且如果执行过 TNS 步骤，就进行时域噪声整形以便控制各转换窗口内量化噪声的瞬时形状。

当量化数据是由符号数据和数值数据组成时，通过顺序解码量化频率分量符号位的数值数据并且耦合数值数据和符号位，恢复量化频率分量。

按照从最高有效位到最低有效位的顺序执行解码步骤，并且通过对已解码的位片式数据耦合以及将耦合数据恢复为量化频率分量数据，执行恢复步骤。

在解码步骤中对以上数据进行解码，从而将四个样本的位片式信息解码为四维向量组。

进行四维向量解码，从而对根据预状态编码的两个子向量进行算术解码，并且将根据各样本的编码状态进行解码的两个子向量恢复为四维向量，其中预状态表示是否对非零位片式频率分量进行过编码。

此外，当根据 MSB 对各频率分量的位片式数据进行解码时，如果位片式数据为“0”就跳过解码，当位片式数据“1”第一次出现时，就对符号数据进行算术解码。通过解码位流中的最大比例因子，对最大比例因子和各比例因子的差值进行算术解码，从最大比例因子中减去该差值，就完成了比例因子的解码。此外，解码比例因子的步骤包括以下步骤：根据位流解码最大比例因子，通过映射获得最大比例因子与需要解码的比例因子之间的差值对该差值进行算术解码，根据映射值，对该差值进行逆映射，通过从最大比例因子中减去该差值获得第一比例因子，以及通过从前一比例因子中减去该差值获得剩余频带的比例因子。

通过执行以下步骤完成算术编码模型系数的解码：解码位流中的最小算术模型系数，解码最小系数与各层边信息中各系数之间的差值，累加最小系数与该差值。

另一方面，根据本发明，提供一种可伸缩的音频解码装置，该装置用于解码音频数据编码以便具有分层位率，该装置包括位流分析部件，用以分析具有分层结构的位流中各模块所需的数据，解码部件，该部件按具有分层结构的位流中各层的建立次序，至少对比例因子、算术编码模型系数和量化数据进行解码，其中按照从较高有效位到较低有效位的顺序，通过分析构成位流的位的有效性，轮流对各声道的量化数据进行解码，恢复部件，该部件将经过解码的比例因子和量化数据恢复为具有原始数值的信号，以及频率/时间映射部件，该部件将量化信号反演为时域信号。

该装置还包括以下部件：进行 M/S 立体声处理的 M/S 立体声处理部件，该部件用于检查是否在位流编码方法中进行过 M/S 立体声处理，并且如果进行过 M/S 立体声处理，就将左声道信号和右声道信号转换为两个信号的相加信号及其相减信号，预测部件，该部件用于检测是否在位流编码方法中执行过预测步骤，并且如果执行过预测步骤，就预测当前帧的频率系数，立体声强度处理部件，该部件用于检测是否在位流编码方法中进行过立体声强度处理，并且如果进行过立体声强度处理，则由于仅对两条声道之一声道（左声道）的比例系数频带的量化信息进行过编码，所以进行立体声强度处理以便将另一声道（右声道）的量化信息恢复到左声道值，和时域噪声整形部件，该部件用于检测是否在位流编码方法中执行过时域噪声整形（TNS）步骤，并且如果执行过 TNS 步骤，就进行瞬时噪声处理以便控制各转换窗口内量化噪声的瞬时形状。

通过参照附图详细说明本发明的最佳实施方式，本发明的上述目的和优点将会更加显而易见。其中附图是：

图 1 是根据本发明的编码装置的框图；

图 2 表示根据本发明的位流的结构；

图 3 是根据本发明的解码装置的框图；

图 4 说明长块（窗口长度=2048）频率分量的排列；和

图 5 说明短块（窗口长度=2048）频率分量的排列。

以下，将参照附图详细说明本发明的最佳实施方式。

本发明准备采用位片式算术编码（BSAC）技术编码并解码可伸缩的立体声数字音频数据。换句话说，在本发明中，仅仅利用 BSAC 技术代替无损编码模块，而并不改变常规编码器的其他模块。本发明推广了按如上方式构造的可伸缩编码器/解码器的适用性，即，可以将本发明应用于立体声信号。

图 1 是根据本发明的可伸缩音频编码装置的框图。可伸缩音频编码装置包括时/频映射部件 100，音质部件 110，时域噪声整形部件 120，立体声强度处理部件 130，预测部件 140，mid/side（M/S）立体声处理部件 150，量化部件 160，位片式算术编码部件 170 和位流形成部件 180。

在对数字音频信号进行编码时，最重要的人类声音特征是屏蔽效果和

临界频带特征。屏蔽效果指由于另一信号影响而听不到某个音频信号（声音）的现象。例如，当一列火车经过某个车站时，受该列火车造成的噪声的影响，在低声谈话期间，他/她不能听到对方的声音。在人类可听到频率范围内，人们对各个频段的音频信号的感受也不尽相同。此外，考虑到临界频带特征，当噪声信号处于临界频带内或噪声信号从关键信号中出来时，人们对具有相同振幅的各种噪声的感受也不相同。此时，当噪声信号越过临界频带时，可以更清楚地听到噪声。

本质上利用以上两个特征对人类音频特征进行编码，从而计算可能位于临界频带内的噪声范围，随后对应于计算范围产生量化噪声，以便将编码造成的信息损失降低到最低限度。

时/频映射部件 100 将时域内的音频信号转换为频域内的音频信号。

音质部件 110 利用预定比例因子频带信号连接时/频映射部件 100 的转换信号，并利用各信号相互作用产生的屏蔽现象计算各比例因子频带的屏蔽阈值。

时域噪声整形部件 120 控制各转换窗口内量化噪声的瞬时形状。通过过滤频率数据，可以对噪声进行瞬时整形。编码器可以选择使用该模块。

立体声强度处理部件 130 是一个模块，该模块用于更有效地处理立体声信号，立体声强度处理部件 130 仅对两个声道中某个声道的比例因子频带的量化信息进行编码，而传输另一个声道的比例因子频带。在编码器中不一定使用该模块，而是需要考虑各比例因子频带的所有要素，以便确定是否使用该模块。

预测部件 140 估计当前帧的频率系数，量化并编码预测值与实际频率分量之间的差值，以便降低生成的可用位的数量。在帧组中选择使用预测部件 140。换句话说，由于使用预测部件 140 增加了预测后继频率系数的复杂性，所以可以不使用预测部件 140。偶尔，经过估计实际生成的位数可能大于不经估计实际生成的位数。此时，不使用预测部件 140。

用于更有效处理立体声信号的 M/S 立体声处理部件 150，将左声道信号和右声道信号分别转换为两个信号的相加信号和相减信号，以便随后处理该信号。在编码器中不一定使用该模块，而是需要考虑各比例因子频带的所有要素，以便确定是否使用该模块。

量化部件 160 标量量化各频带的频率信号，从而各频带量化噪声的数值小于屏蔽阈值，以致觉察不到量化噪声。进行量化以使 NMR（噪声屏蔽率）值小于等于 0 dB，其中 NMR 为音质部件 210 计算的屏蔽阈值与各频带所产生的噪声的比率。NMR 值小于等于 0 dB 意味着屏蔽阈值大于量化噪声。换句话说，听不到量化噪声。

由于诸如 MPEG-2 ACC 之类的现有音频编码解码器不具有可伸缩性，所以可以使用本发明的核心模块，即位片式算术编码部件 170，作为 ACC 技术的无损编码部件的备择方案。为了实现可伸缩的音频编码解码器，通过组合相应频带的边信息和音频数据的量化数据，对量化部件 160 产生量化的频率数据进行编码。此外，除可伸缩性之外，可以在顶层提供 ACC 中的类似功能。以下详细说明位片式算术编码部件 170 的功能。将频带限制为与基准层相对应的频带以便伸缩，并对基准层的边信息进行编码。按照从 MSB 顺序到 LSB 顺序，从低频分量到高频分量的次序，对量化值信息进行顺序编码。此外，以预定向量组的方式对左声道和右声道轮流编码，以便对基准层进行编码。在结束基准层编码后，对下一增强层的边信息以及音频数据的量化值进行编码，从而按上述方式构成的位流具有分层结构。

位流形成部件 180 通过收集编码装置的各模块内形成的信息，根据适用于可伸缩编码解码器的预定语法产生位流。

图 2 表示根据本发明的位流的结构。如图 2 所示，该位流具有分层结构，其中根据位率，较低位率层的位流包含在较高位率层的位流中。通常，首先编码边信息，随后编码剩余信息，从而形成位流。然而，如图 2 所示，在本发明中分别编码各增强层的边信息。此外，尽管通常以采样组合的方式顺序编码量化数据，在本发明中，利用二进制数据表示量化数据，并且根据二进制数据的 MSB 顺序编码量化数据，以便在指定位内形成位流。

图 3 是根据本发明的解码装置的框图，解码装置包括位流分析部件 300，位片式算术解码部件 310，反向量化部件 320，M/S 立体声处理部件 330，预测部件 340，立体声强度处理部件 350，时域噪声整形部件 360，和频率/时间映射部件 370。

位流分析部件 300 以输入位流的生成次序分离标题信息和编码数据，

并将其传输到各模块。

位片式算术解码部件 310 以输入位流的生成次序解码边信息和位片式量化数据，并将解码结果传输到反向量化部件 320。

仅接受立体声信号的 M/S 立体声处理部件 330，处理与编码装置中进行 M/S 立体声处理相对应的比例因子频带。

当在编码装置中进行估计时，与编码装置中采用的方法相同，预测部件 340 以前一帧中解码数据的方式，经过估计搜索同一值。将位流分析部件 300 解码的差分信号加入到预测信号中，从而恢复原始频率分量。

仅接受立体声信号的立体声强度处理部件 350，处理与编码装置中进行 M/S 立体声处理相对应的比例因子频带。

用于控制各转换窗口内量化噪声瞬时形状的时域噪声整形部件 360 进行相应的处理。

采用诸如 ACC 标准件之类的常规音频算法处理模块，将经过解码的数据恢复为时域信号。首先，反向量化部件 320 将经过解码的比例因子和量化数据恢复为具有原始数值的信号。随后，频率/时间映射部件 370 将量化信号反演为时域信号，以便还原。

现在将说明编码装置的操作。

在时间/频率映射部件 100，经过 MDCT（改良离散余弦变换），将输入信号转换为频域信号。音质部件 110 利用适宜的比例因子频带连接以上频域信号，以便获得屏蔽阈值。此外，转换为频域信号的音频信号通过用于提高编码效率的组件，即 TNS 部件 120，立体声强度处理部件 130，预测部件 140 和立体声处理部件 150，以便更有效地压缩信号。

量化部件 160 进行标量量化，从而各比例因子频带的量化噪声值小于屏蔽阈值，可以听到但不会在指定位内觉察到该值。如果进行了满足以上条件的量化，就产生各比例因子频带的比例因子和量化频率值。

通常，考虑到人类音质，可以在较低的频率轻易觉察到几乎相等的频率分量。然而，随着频率的增加，可觉察的频率区间变宽。比例因子频带的带宽随着频带的增高而增高。然而，为了方便编码，并不采用其带宽不固定的比例因子频带进行编码，而是采用其带宽固定的编码频带。编码频带包括 32 个经过量化的频率系数值。

在处理立体声信号时，诸如 ACC 之类的仅考虑编码效率的常规编码/解码装置，首先对左右声道共同使用的信息进行编码。编码左声道数据，随后编码右声道数据。即以标题、左声道和右声道的次序进行编码。

如上所述，在处理了标题之后，当不考虑其有效值而排列并传输左右声道信息时，如果位率下降，则位于后边的右声道信号首先消失。因此，可显著觉察到性能下降。

然而，根据本发明的音频编码装置对各声道的边信息进行编码。换句话说，位片式算术编码部件 170 以左声道和右声道的次序，轮流编码各声道的边信息。为了有效压缩，稍微改进比例因子编码方法。

首先说明比例因子编码。根据本发明的立体声音频编码装置采用将在下面说明的两种方法对比例因子进行编码，以便提高编码效率。编码装置选择呈现较好性能的方法，并且将所选方法传输到解码装置。

第一，为了压缩比例因子，根据所有比例因子获得最大比例因子 (max_scalefactor)。然后，获得各比例因子与最大比例因子之间的差值，并对该差值进行算术编码。采用四个模型对比例因子之间的差值进行算术编码。表 5.5 到表 5.8 表示以上四个模型。在 scalefactor_model 中存储模型信息。

[表 5.5]差分比例因子算术模型 1

长度	累加频率
8	1342,790,510,344,214,127,57,0

[表 5.6]差分比例因子算术模型 2

长度	累加频率
16	2441,2094,1798,1563,1347,1154,956,818,634,464,342,241,157,97,55,0

[表 5.7]差分比例因子算术模型 3

长度	累加频率
32	3963,3525,3188,2949,2705,2502,2286,2085,1868,1668,

	1515,1354,1207,1055,930,821,651,510,373,269,192, 134,90,58,37,29,24,15,10,8,5,0
--	------------------------------------------------------------------------------------

[表 5.8]差分比例因子算术模型 4

长度	累加频率
64	13587,13282,12961,12656,12165,11721,11250,10582, 10042,9587,8742,8010,7256,6619,6042,5480,4898,4331, 3817,3374,3058,2759,2545,2363,2192,1989,1812,1582, 1390,1165,1037,935,668,518,438,358,245,197,181,149, 144,128,122,117,112,106,101,85,80,74,69,64,58,53, 48,42,37,32,26,21,16,10,5,0

第二，为了压缩比例因子，正如第一方法，根据所有比例因子获得最大比例因子（`max_scalefactor`）。然后，获得第一比例因子与最大比例因子之间的差值，并对该差值进行算术编码。然后，获得剩余比例因子与先前比例因子之间的差值，并对所有差值进行算术编码。这样，由于指定了所用模型，所以 `scalefactor_model` 的值是无意义的。

下面，将说明对立体声信号量化频率分量进行的编码。各频道的量化数据是位片式的。当处理单声道信号时，利用四维向量耦合位片式数据，并且将四维向量用作基本组件。这种处理同样适用于对立体声声道信号的编码。换句话说，从 MSB 开始编码。从左声道开始对位片式数据的四维向量进行算术编码。接着，对位于相同频率的右声道的四维向量进行算术编码。这样，对左声道和右声道交替进行编码。

就单声道而论，按照从 MSB 到 LSB 的次序进行编码。按照从较低频率分量到较高频率分量的次序，对具有相同有效值的位片式数据进行编码。此时，如果分配到各向量的所有位比当前正被编码的位更有效，则不必对各向量进行编码，并跳过该编码处理。

`XQ0, XQ1, XQ2, ..., XQk, ...`

其中 `Xqk` 为从 $4*k$ 到 $4*k+3$ 的量化频率分量的位片式数据。

就双声道而言，如同单声道一样，按照从 MSB 到 LSB 的次序进行编码。同样，按照从较低频率分量到较高频率分量的次序，对具有相同有效值的位片式数据进行编码。然而，考虑到存在两个声道，需要确定编码顺序。假设左声道和右声道中的量化频率分量如下：

左声道： $XQL0, XQL1, XQL2, XQL3, XQL4, XQL5, \dots, XQLk, \dots$

右声道： $XQR0, XQR1, XQR2, XQR3, XQR4, XQR5, \dots, XQRk, \dots$

其中 $XQLk$ 和 $XQLRk$ 为从 $4*k$ 到 $(4*k+3)$ 的量化频率分量的位片式数据。

这样，就双声道而言，以单声道中类似次序的方式，按照从较低频率分量到较高频率分量的次序进行编码。然而，为了首先对有效分量进行编码，需要交错声道分量。换句话说，按如下方式轮流对两个声道之间的各向量进行编码。

$XQL1, XQR1, XQL2, XQR2, \dots$

由于按照两个声道内有效值的次序，顺序对按上述方式构成的信息进行编码，所以即使降低可伸缩音频编码解码器的位率，其性能也不会显著下降。

现在，将说明本发明的最佳实施方式。本发明沿用 ACC 标准件的基本结构实现可伸缩的数字音频数据编码器，其中 ACC 标准件包括诸如附加模块（该模块用于提高编码效率）之类的所有模块。换句话说，在本发明中，在采用 ACC 标准编码/解码件中基本模块的同时，利用位片式编码方法取代无损编码模块，以便提供可伸缩的编码装置。在本发明中，在单一位流内不对仅为某一位率的信息进行编码，而在具有分层结构的位流（如图 2 所示）内，按照从较重要的信号分量到不太重要的信号分量的次序，对不同增强层的位率信息进行编码。

根据本发明的实施方式，在 BSAC 可伸缩编码解码器的无损编码之前，采用与 ACC 标准件相同的模块。因此，如果量化频率数据是通过 ACC 位流进行解码而形成的，则可以将经过解码的数据恢复为 BSAC 可伸缩位流。这意味着可以在 ACC 位流和 BSAC 可伸缩位流之间进行自动译码。最后，根据环境提供与位流格式相称的相互转换。因此，可以同时满足编码效率和可伸缩性，并且编码效率和可伸缩性是互补的，这正是与其

他可伸缩的编码解码器的区别。

利用按上述方式构成的位流，根据用户请求或传输声道的状态，通过简单重排包含在最高位流中的低位率位流，就能够形成具有低位率的位流。换句话说，可以将编码装置采用实时算法形成的位流或者存储在媒体上的位流，重排为适于用户请求的理想位率，以便传输。此外，即使具有适宜位流，如果用户的硬件性能较差，或者用户希望降低解码器的复杂性，也仅能恢复某些位流，从而控制了复杂性。

例如，在形成可伸缩位流时，基准层的位率为 16 Kbps，顶层的位率为 64 Kbps，各增强层的位率间隔为 8 Kbps，即该位流具有 7 层，各层位率分别为 16、24、32、40、48、56 和 64 Kbps。按照表 2.1 所示定义各增强层。如图 3 所示，由于编码装置形成的位流具有分层结构，所以 64 Kbps 的顶层位流包含各增强层的位流（16、24、32、40、48、56 和 64 Kbps）。如果用户请求顶层数据，则不进行任何处理就传输顶层位流。此外，如果另一用户请求基准层数据（对应于 16 Kbps），则只传输前导位流。

[表 2.1]各层的位率（8 kbps 间隔）

层	位率（kbps）
0	16
1	24
2	32
3	40
4	48
5	56
6	64

另一方面，可以以更细的间隔构造增强层。基准层的位率为 16 Kbps，顶层的位率为 64 Kbps，各增强层的位率间隔为 1 Kbps。按照表 3.1 所示构造增强层。因此，可以实现精粒可伸缩性，即以 1 kbps 的位率间隔，形成从 16 kbps 到 64 kbps 的可伸缩位流。

[表 3.1]各层的位率（1-kbps 间隔）

层	位率	层	位率	层	位率	层	位率
0	16	12	28	24	40	36	52
1	17	13	29	25	41	37	53
2	18	14	30	26	42	38	54
3	19	15	31	27	43	39	55
4	20	16	32	28	44	40	56
5	21	17	33	29	45	41	57
6	22	18	34	30	46	42	58
7	23	19	35	31	47	43	59
8	24	20	36	32	48	44	60
9	25	21	37	33	49	45	61
10	26	22	38	34	50	46	62
11	27	23	39	35	51	47	63
						48	64

根据位率，各层具有有限带宽。如果想要 8 kbps 间隔的可伸缩性，则按照表 2.2 和表 2.3 所示限制带宽。对于 1 kbps 间隔而言，按照表 3.2 和表 3.3 所示限制带宽。

[表 2.2]用于短窗口的各层内的带限（8-kbps 间隔）

层	带限
0	20
1	28
2	40
3	52
4	60
5	72
6	84

[表 2.3]用于长窗口的各层内的带限（8-kbps 间隔）

层	带限
---	----

0	160
1	244
2	328
3	416
4	500
5	584
6	672

[表 3.2]用于短窗口的各层内的带限 (1-kbps 间隔)

层	带限	层	带限	层	带限	层	带限
0	20	12	36	24	52	36	68
1	20	13	36	25	52	37	68
2	20	14	36	26	52	38	68
3	24	15	40	27	56	39	72
4	24	16	40	28	56	40	72
5	24	17	40	29	56	41	72
6	28	18	44	30	60	42	76
7	28	19	44	31	60	43	76
8	28	20	44	32	60	44	76
9	32	21	48	33	64	45	80
10	32	22	48	34	64	46	80
11	32	23	48	35	64	47	80
						48	84

[表 3.3]用于长窗口的各层内的带限 (1-kbps 间隔)

层	带限	层	带限	层	带限	层	带限
0	160	12	288	24	416	36	544
1	168	13	296	25	424	37	552
2	180	14	308	26	436	38	564

3	192	15	320	27	448	39	576
4	200	16	328	28	456	40	584
5	212	17	340	29	468	41	596
6	224	18	352	30	480	42	608
7	232	19	360	31	488	43	616
8	244	20	372	32	500	44	628
9	256	21	384	33	512	45	640
10	264	22	392	34	520	46	648
11	276	23	404	35	532	47	660
						48	672

输入数据为以 48 KHz 频率进行采样的 PCM 数据，并且每帧的数据量为 1024 。就 64 Kbps 的位率而言，一帧内的平均可用位数为 $1365.3333(=64000 \text{ 位/秒} \times (1024/48000))$ 。同样，根据各位率，就可以计算各帧的可用位数。就 8 Kbps 的位率而言，表 2.4 说明一帧内所有可用位的计算数，就 1 Kbps 的位率而言，表 3.4 说明一帧内所有可用位的计算数。

[表 2.4]各层内各声道的可用位（8-kbps 间隔）

层	可用位
0	341
1	512
2	682
3	853
4	1024
5	1194
6	1365

现在，将详细说明根据本发明的立体声音频信号的编码和解码过程。

1. 编码过程

整个编码过程与 MPEG-2 ACC 国际标准中的叙述相同，并且采用本发明推荐的位片式编码作为无损编码。

1.1. 音质部件

利用音质模型，根据输入数据，首先生成正在处理的帧的块类型（长、启动、短、停止），各处理频带的 SMR 值，短块的分组信息，以及用于采用音质模型时/频同步的瞬时延迟 PCM 数据，并将其传输到时间/频率映射部件。采用 ISO/IEC 11172-3 模型 2 计算音质模型 [MPEG 委员会 ISO/IEC/JTC1/SC29/WG11，信息技术-以大约 1.5 Mbit/s 速率对用于数据存储介质的动画以及各音频信号进行编码 - 第三部分：音频，ISO/OEC IS 11172-3，1993]。必须使用该模块，但是可以根据用户要求采用不同的模型。

1.2. 时间/频率映射部件

采用 MPEG-2 ACC 国际标准中定义的时间/频率映射部件。利用音质模型，根据块类型输出，时间/频率映射部件采用 MDCT 将时域数据转换位频域数据。此时，关于长/启动/停止块和短块而言，其块长度分别为 2048 和 256，并且执行 8 次 MDCT。然后，将窗口类型和窗口分组信息传输到位流形成部件 180。迄今为止，一直采用常规 MPEG-2 ACC [MPEG 委员会 ISO/IEC/JTC1/SC29/WG11，ISO/IEC MPEG-2 ACC IS 13818-7，1997] 中所采用的相同过程。

1.3. 时域噪声整形部件（TNS）

采用 MPEG-2 ACC 国际标准中定义的时域噪声整形部件。TNS 120 是一个选件，该选件控制各转换窗口内量化噪声的瞬时形状。通过过滤频率数据，就能够实现时域噪声整形。TNS 120 将 TNS 信息传输到位流形成部件 180。

1.4. 立体声强度处理部件

采用 MPEG-2 ACC 国际标准中定义的立体声强度处理部件。立体声强度处理部件 130 是一种更有效处理立体声信号的方法。进行立体声强度处理，从而仅对两条声道中某条声道的比例因子频带的量化信息进行编码，而仅仅传输另一条声道的比例因子。该模块是可选模块，并且考虑各种条件，以便确定各比例因子频带是否使用该模块。立体声强度处理部件 130 将立体声强度标志值传输到位流形成部件 180。

1.5. 预测部件

采用 MPEG-2 ACC 国际标准中定义的预测部件。预测部件 140 为可选模块，该模块预测当前帧的频率系数。此外，预测部件 140 将和预测各的参数传输到位流形成部件 180。

1.6.Mid/Side (M/S)立体声处理部件

采用 MPEG-2 ACC 国际标准中定义的 M/S 立体声处理部件。M/S 立体声处理部件 150 为可选模块，该模块是一种更有效处理立体声信号的方法。进行 M/S 立体声处理以便将左声道信号和右声道信号转换为两个信号的相加信号以及其相减信号。

1.7.量化部件

利用渐增比例因子对转换为频域的数据进行量化，从而表 1.1 和表 1.2 所示的比例因子频带的 SNR 值小于音质模型的输出值 SMR。这里，进行标量量化，并且基础比例因子为 $2^{1/4}$ 。进行量化，从而将可觉察噪声降到最低限度。MPEG-2 ACC 中说明了精确的量化过程。这里，得到的输出为各比例因子频带的量化数据和比例因子。

[表 1.1]长块的比例因子频带

swb	swb_offset_ long window	swb	swb_offset_ long window	swb	swb_offset_ long window	swb	swb_offset_ long window
0	0	12	56	24	196	37	576
1	4	13	64	25	216	38	608
2	8	14	72	26	240	39	640
3	12	15	80	27	264	40	672
4	16	16	88	28	292	41	704
5	20	17	96	29	320	42	736
6	24	18	108	30	352	43	768
7	28	19	120	31	384	44	800
8	32	20	132	32	416	45	832
9	36	21	144	33	448	46	864
10	40	22	160	34	480	47	896

11	48	23	176	35	512	48	928
				36	544		1024

[表 1.2]短块的比例因子频带

swb	swb_offset_shortwindow	swb	swb_offset_shortwindow
0	0	8	44
1	4	9	56
2	8	10	68
3	12	11	80
4	16	12	96
5	20	13	112
6	28		128
7	36		

1.8.利用位片式算术编码进行位压缩

利用位片式算术编码部件 170 和位流形成部件 180 进行位压缩。为了方便编码，重新排列频率分量。取决于块类型，重排次序并不相同。如图 4 所示，当在块类型中使用长窗口时，按照比例因子频带的次序重排频率分量。如图 5 所示，当在块类型中使用短窗口时，按照渐增次序重复排列 8 块中的各个四频率分量。

经过重排的量化数据和比例因子构成分层位流。利用表 7.1 到表 7.3 所示的句法构造位流。位流的前导元素为常规 ACC 中所使用的元素，并且对本发明新推荐的元素进行具体解释。然而，其基本结构与 ACC 标准类似。

[表 7.1]bsac_1step_data_block()句法

句法	位数	助记符
<pre>bsac_1step_data_block() { lslayer=0; while(data_available()){ bsac_1step_stream(lslayer) } lslayer++; }</pre>		

[表 7.2]bsac_1step_stream()句法

句法	位数	助记符
<pre>bsac_1step_stream(lslayer) { for(i=lstep_offset[lslayer]; i<lstep_offset[lslayer+1]; i++) BSAC_stream_buf[i] /*将长步流保存在 BSAC_stream_buf[] 中。 将 BSAC_stream_buf[]映射到小步流, bsac_raw_data_block(), 以便实际解 码。 详细说明参见 BSAC 长步可伸缩性的解 码过程。 */ }</pre>	8	unimsbf

[表 7.3]bsac_raw_data_block()句法

句法	位数	助记符
----	----	-----

<pre> bsac_raw_data_block() { bsac_main_stream() layer=1; while(data_available() layer<=encoded_layer){ bsac_layer_stream(nch, layer) layer++; } byte_alignment() } </pre>		
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

[表 7.4]bsac_main_stream()句法

句法	位数	助记符
<pre> bsac_main_stream() { nch switch(nch){ case 1: bsac_single_main_stream() break case 2: bsac_pair_main_stream() break } } </pre>	3	unimsbf

[表 7.5]bsac_single_main_stream()句法

句法	位数	助记符
----	----	-----

}		
---	--	--

[表 7.8]bsac_channel_stream()句法

句法	位数	助记符
bsac_channel_stream(nch,common_window)		
{		
for(ch=0;ch<nch;ch++)	8	uimbf
max_scalefactor[ch]		
ics_info()		
if(!common_window)		
ics_info()		
for(ch=0;ch<nch;ch++){	1	uimbf
tns_data_present[ch]		
if(tns_data_present[ch])		
tns_data()	1	uimbf
gain_control_data_present[ch]		
if(gain_control_data_present[ch])		
gain_control_data()		
}	1	uimbf
PNS_data_present		
if(PNS_data_present)	6	uimbf
PNS_start_sfb		
bsac_general_info(nch)		

bsac_layer_stream(nch,0) }		
---------------------------------------------	--	--

[表 7.9]bsac_general_info()句法

句法	位数	助记符
bsac_general_info(nch) { frame_length encoded_layer for(ch=0;ch<nch;ch++){ scalefactor_model[ch] main_ArModel[ch] ArModel_model[ch] scf_coding[ch] } }	 10/11 6 2 5 2 1	 uimbf uimbf uimbf uimbf uimbf uimbf

[表 7.10]bsac_side_info()句法

句法	位数	助记符
bsac_side_info(nch,layer) { if(nch==1 && PNS_data_present){ for(sfb=PNS_start_sfb; sfb<max_sfb; sfb++) acode_noise_flag[g][sfb] } else if(stereo_mode > 1 PNS_data_present) for(g=0;g<num_window_group;g++) for(sfb=layer_sfb[layer];sfb<layer_sfb[layer+ 1];sfb++)	 0..1	 bslbf

<pre> { if(stereo_mode == 2) acode_ms_used[g][sfb] else if(stereo_mode == 3) acode_stereo_info[g][sfb] } </pre>	0..1	bslbf
<pre> if(PNS_data_present sfb>=PNS_start_sfb){ </pre>	&&	bslbf
<pre> if(stereo_info == 0 stereo_info == 3){ acode_noise_flag_l[g][sfb] acode_noise_flag_r[g][sfb] } </pre>	0..1	bslbf
<pre> if(stereo_info == 3){ if(noise_flag_l[g][sfb] noise_flag_r[g][sfb]) acode_noise_mode[g][sfb] } } } } </pre>	0..1	bslbf
<pre> for(ch=0; ch<nch; ch++) for(g=0; g<num_window_group; g++) for(sfb=layer_sfb[layer]; sfb<layer_sfb[layer+1]; sfb++) acode_scf[ch][g][sfb] for(ch=0; ch<nch; ch++) for(sfb=layer_sfb[layer]; sfb<layer_sfb[layer+1]; sfb++) </pre>	0..2	bslbf
	&&	bslbf
	0..13	bslbf

<pre> for(g=0; g<num_window_group; g++){ band=(sfb*num_window_group) + g for(i=swb_offset[band]; i<swb_offset[band+1]; i+=4) cband = index2cb(ch,i); if(!decode_cband[ch][cband]){ acode_ArMode[ch][cband] decode_cband[ch][cband] = 1; } } } } </pre>	0..13	bslbf
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------	-------

[表 7.11]bsac_spectral_data()句法

句法	位数	助记符
<pre> bsac_spectral_data(nch,layer) { for(snf=maxsnf; snf>0; snf--){ for(i=0; i<last_index; i+=4){ for(ch=0; ch<nch; ch++){ if(i>=layer_index[ch]) continue; if(cur_snf[ch][i]<snf) continue; dim0 = dim1 = 0 for(k = 0; k < 4; k++) if(prestate[ch][i + k]) dim1++ else dim0++ if(dim0) acode_vec0 if(dim1) </pre>	0.14	bslbf

<pre> acode_vec1 for(k = 0; k < 4; k++) if(sample[ch][i+k] && !prestate[ch][i+k]){ acode_sign prestate[ch][i+k] = 1 } } cur_snf[ch][i]-- if(total_estimated_bits >= available_bits[layer]) return } } if(total_estimated_bits >= available_bits[layer]) return } } </pre>	0.14	bslbf
	1	bslbf

以下将具体解释本发明新推荐的元素。

1.8.1.bsac_channel_stream 的编码

‘common_window’ 表示两个声道是否使用相同的格式块，
‘max_scalefactor[ch]’ 表示比例因子的最大值，该值为整数，如 8 位。
此外，‘tns_data_present[ch]’ 表示是否在编码装置中采用 TNS。
‘gain_control_data_present[ch]’ 表示一个标志，该标志表明为了在 ACC
中提供可伸缩的采样率（SSR）而采用的时间/频率映射方法。此外，
‘stereo_mode’ 表示一个 2 位标志，该标志说明立体声信号处理方法，
其中 ‘00’ 表示独立，‘01’ 表示所有 ms_used 均为 1，‘10’ 表示在
该层的边信息中存储有 ms_used 的 max_sfb 频带的 1 位屏蔽，‘11’ 表
示在该层的边信息中存储有 stereo_info 的 max_sfb 频带的 2 位屏蔽。

1.8.2.bsac_data 的编码

‘frame_length’表示某帧内所有位流的长度，其单位为字节，例如，对于单声道信号（M/S）而言，其长度为9位，对于立体声信号而言，其长度为10位。此外，‘encoded_layer’表示在位流内编码的顶层的编码，对于8-kbps间隔而言该编码为3位，而对于1-kbps间隔而言该编码为6位。表2.1和表3.1表示增强层的信息。此外，‘scalefactor_model[ch]’表示与模型各的信息，该模型用于对比例因子之间的差值进行算术编码。表4.2表示该模型。

[表 4.2]差分比例因子的算术模型

模型号	最大差分比例因子	模型列表
0	7	表 5.5
1	15	表 5.6
2	31	表 5.7
3	63	表 5.8

‘min_ArModel’表示算术编码模型系数的最小值。

‘ArModel_model’表示与模型各的信息，该模型用于对ArModel和min_ArModel之间的差分信号进行算术编码。表4.3表示该信息。

[表 4.3]差分ArModel的算术模型

模型号	最大差分比例因子	模型列表
0	3	表 5.9
1	7	表 5.10
2	15	表 5.11
3	31	表 5.12

1.8.3.bsac_side_info 的编码

首先对能够用于所有层的信息进行编码，随后对通常用于各增强层的边信息进行编码。‘acode_ms_used[g][sfb]’表示对ms_used进行算术编码得到的编码字，ms_used为1位标志，该标志表示是否在窗口分组g和比例因子频带scf内进行M/S编码，其中ms_used定义如下：

0: 独立

1: ms_used.

‘acode_ms_used[g][sfb]’表示对ms_used进行算术编码得到的编码字，ms_used为1位标志，该标志表示是否在窗口分组g和比例因子频带scf内采用M/S编码，其中ms_used定义如下：

0: 独立；和

1: ms_used.

‘acode_stereo_info[g][sfb]’表示对ms_used进行算术编码得到的编码字，ms_used为2位标志，该标志表示是否在窗口分组g和比例因子频带scf内采用立体声强度编码，其中stereo_info定义如下：

00: 独立；

01: ms_used；

10: Intensity_in_phase；和

11: Intensity_out_of_phase.

‘Acode_scf’表示对比例因子进行算术编码得到的编码字，‘acode_ArModel’表示对ArModel进行算术编码得到的编码字。ArModel为根据表4.3所列模型选择的信息。

1.8.4.bsac_spectral_data的编码

利用BSAC技术，对通常用于各增强层的边信息以及经过量化的频率分量进行位片式处理，随后对其进行算术编码。‘acode_vec0’表示利用等于ArModel值的算术模型对第一子向量(subvector 0)进行算术编码而得到的编码字。‘acode_vec1’表示利用等于ArModel值的算术模型对第二子向量(subvector 1)进行算术编码而得到的编码字。‘acode_sign’表示利用表5.15定义的算术模型对符号位进行算术编码而得到的编码字。

[表 5.15]符号算术模型

长度	累加频率
2	8192, 0

在计算对各子向量进行编码所使用的位数，并与各增强层的可用位数进行比较时，当使用位数大于等于可用位数时，才开始对下一增强层进行编码。

就长块而言，基准层的带宽被限制在第 21 条比例因子频带之下。然后，对前 21 条比例因子频带的比例因子，以及相应编码频带的算术编码模型进行编码。根据算术编码模型得到位分配信息。根据分配到各编码频带的位信息得到所分配位的最大值，并且利用前述编码方法，从最大量化位值开始进行编码。然后，顺序对下一量化位进行编码。如果某频带所分配的位数小于正被编码的频带所分配的位数，则不进行编码。如果某频带所分配的位数等于正被编码的频带所分配的位数，则首次对该频带进行编码。由于基准层的位率为 16 Kbps，所以全部位容量为 336 位。因此，连续计算总使用位数，并且当位数超过 336 位时，就终止编码。

在形成基准层（16 Kbps）的所有位流后，就形成下一增强层的位流。由于增加了较高层的极限带宽，所以仅对基准层极限频带的新增频带的比例因子和算术编码模型进行编码。在基准层，按照基准层内的相同方式，从 MSB 开始，对各频带未编码的位片式数据以及新增频带的位片式数据进行编码。当总使用位数大于可用位数时，就终止编码，并且准备形成下一增强层位流。这样，就能够生成 32、40、48、56 和 64 Kbps 的剩余层的位流。

2. 解码过程

2.1. 位流的分析和解码

2.1.1. bsac_channel_stream 的解码

按照以下次序，对 bsac_channel_stream 进行解码。首先，得到 max_scalefactor。然后，得到 ics_info()。如果有 TNS 数据，则得到 TNS 数据。如果有两个声道，则得到 stereo_mode，随后得到 BSAC 数据。

2.1.2. bsac_data 的解码

在位流内，对解码 frame_length 所需的边信息、encoded_layer、比例因子模型以及算术模型进行解码。

2.1.3. bsac_stream 的解码

BSAC 流具有分层结构。首先，从位流中分离出基准层的边信息，随后对其进行算术解码。然后，从位流中分离出经过量化的频率分量的位片式信息，并对其进行算术解码。接着，对下一增强层的边信息进行解码，并对经过量化的频率分量的位片式信息进行算术解码。

反复对各增强层的边信息以及位片式数据进行解码，直至增强层大于编码层。

2.1.4.stereo_info 或 ms_used 的解码

对 stereo_info 或 ms_used 进行的解码受 stereo_mode 的影响，其中 stereo_mode 表示立体声屏蔽。如果 stereo_mode 为 0 或 1，则无需对 stereo_info 或 ms_used 进行解码。

如果 stereo_mode 为 1，则所有的 ms_used 均为 1。将有关 ms_used 的信息传输到 M/S 立体声处理部件，从而开始 M/S 立体声处理。如果 stereo_mode 为 2，则采用表 5.13 所示的模型对 ms_used 的值进行算术编码。此外，将有关 ms_used 的信息传输到 M/S 立体声处理部件，从而开始 M/S 立体声处理。

[表 5.13]ms_used 模型

长度	累加频率
2	11469, 0

如果 stereo_mode 为 3，则采用表 5.14 所示的模型对 stereo_info 进行算术编码。正如 ACC 中所述，将经过解码的数据传输到 M/S 立体声处理部件或者立体声强度处理部件，从而以比例因子频带组的形式开始 M/S 立体声处理或者立体声强度处理。

[表 5.14]stereo_info 模型

长度	累加频率
2	13926, 4096, 1638, 0

2.1.5bsac_side_info 的解码

按上述方式形成的可伸缩位流具有分层结构。首先，从位流中分离出基准层的边信息，随后对其进行解码。然后，从位流中分离出包含在基准层位流中的经过量化的频率分量的位片式信息，并对其进行解码。对其他增强层施加与基准层相同的解码过程。

2.1.5.1.比例因子的解码

将频率分量划分为比例因子频带，其中该比例因子频带的频率系数为 4 的倍数。每条比例因子频带均具有一个比例因子。采用两种方法对比例因

子进行解码。根据 scf_coding 的值确定采用的方法。

首先, 将 max_scalefactor 解码为 8 位无符号整数。通常, 在编码期间, 对经过映射差值而得到的值进行编码。因此, 对于各比例因子频带, 采用表 5.2 所示的模型对经过映射的值进行算术编码。此时, 如果经过算术解码的值为 54, 这意味着映射值大于等于 54, 由于还对 54 与映射值之间的差值进行过编码, 所以还需要对经过编码的差值进行解码, 以便将其恢复为大于等于 54 的值。如果完成了对映射值的解码, 就利用差分符号实现映射值的逆映射。利用表 5.1 和表 5.2 所示的映射表进行映射和逆映射。利用 max_scalefactor 与第一比例因子之间的差分符号, 就可以得到第一比例因子。

表 5.1. 从差分比例因子到系数的转换表

D	I	D	I	D	I	D	I	D	I	D	I	D	I	D	I
0	68	16	87	32	46	48	25	64	9	80	40	96	96	112	112
1	69	17	88	33	47	49	19	65	10	81	43	97	97	113	113
2	70	18	89	34	48	50	20	66	12	82	44	98	98	114	114
3	71	19	72	35	49	51	14	67	13	83	45	99	99	115	115
4	75	20	90	36	50	52	15	68	17	84	52	100	100	116	116
5	76	21	73	37	51	53	16	69	18	85	53	101	101	117	117
6	77	22	65	38	41	54	11	70	21	86	63	102	102	118	118
7	78	23	66	39	42	55	7	71	22	87	56	103	103	119	119
8	79	24	58	40	35	56	8	72	26	88	64	104	104	120	120
9	80	25	67	41	36	57	5	73	27	89	57	105	105	121	121
10	81	26	59	42	37	58	2	74	28	90	74	106	106	122	122
11	82	27	60	43	29	59	1	75	31	91	91	107	107	123	123
12	83	28	61	44	38	60	0	76	32	92	92	108	108	124	124
13	84	29	62	45	30	61	3	77	33	93	93	109	109	125	125
14	85	30	54	46	23	62	4	78	34	94	94	110	110	126	126
15	86	31	55	47	24	63	6	79	39	95	95	111	111	127	127

表 5.2.从系数到差分比例因子的转换表

I	D	I	D	I	D	I	D	I	D	I	D	I	D	I	D
0	60	16	53	32	76	48	34	64	88	80	9	96	96	112	112
1	59	17	68	33	77	49	35	65	22	81	10	97	97	113	113
2	58	18	69	34	78	50	36	66	23	82	11	98	98	114	114
3	61	19	49	35	40	51	37	67	25	83	12	99	99	115	115
4	62	20	50	36	41	52	84	68	0	84	13	100	100	116	116
5	57	21	70	37	42	53	85	69	1	85	14	101	101	117	117
6	63	22	71	38	44	54	30	70	2	86	15	102	102	118	118
7	55	23	46	39	79	55	31	71	3	87	16	103	103	119	119
8	56	24	47	40	80	56	87	72	19	88	17	104	104	120	120
9	64	25	48	41	38	57	89	73	21	89	18	105	105	121	121
10	65	26	72	42	39	58	24	74	90	90	20	106	106	122	122
11	54	27	73	43	81	59	26	75	4	91	91	107	107	123	123
12	66	28	74	44	82	60	27	76	5	92	92	108	108	124	124
13	67	29	43	45	83	61	28	77	6	93	93	109	109	125	125
14	51	30	45	46	32	62	29	78	7	94	94	110	110	126	126
15	52	31	75	47	33	63	86	79	8	95	95	111	111	127	127

第二，将 `max_scalefactor` 解码为 8 位无符号整数。对于所有的比例因子而言，对偏移值（即 `max_scalefactor`）与所有比例因子之间的差值进行算术编码。从 `max_scalefactor` 中减去差值符号就能够得到比例因子。对以上差值进行解码的算术模型为构成位流的元素之一，并且该模型是从已经解码的位流中分离出来的。

以下伪码说明对基准层以及其他增强层的比例因子进行解码的方法。

```

for(ch=0; ch<nch; ch++)
    if(scfcoding[ch]==1)
        for(g=0; g<num_window_group; g++)
            for(sfb=layer_sfb[layer]; sfb<layer_sfb[layer+1]; sfb++){
                sf[ch][g][sfb] = max_scalefactor - arithmetic_decoding();
            }
    
```

```

    }
    }
}
else{
    for(g=0; g<num_window_group; g++){
        for( sfb=layer_sfb[layer]; sfb<layer_sfb[layer+1]; sfb++)
            tmp_index = arithmetic_decoding();
        if(tmp_index == 54)
            tmp_index = 54 + arithmetic_decoding();
        if(sfb == 0)
            tmp_index = max_scalefactor - tmp_index;
        else
            tmp_index = sf[ch][g][sfb-1] - tmp_index;
        sf[ch][g][sfb] = index2sf[tmp_index];
    }
}
}
}

```

这里，`layer_sfb[layer]`是对各增强层中的比例因子进行解码所需的起始比例因子频带，而`layer_sfb[layer+1]`是结束比例因子频带。

2.1.5.2. 算术模型系数的解码

将频率分量划分为具有 32 个频率系数的编码频带，以便进行无损编码。编码频带是无损编码中使用的基本部件。

算术编码模型系数是与模型各的信息，其中该模型用于对各频带的位片式数据进行算术编码/解码，该系数表示在算术编码/解码过程中采用表 4.4 所列模型中的哪个模型。

[表 4.4]BSAC 算术模型参数

ArModel 索引	编码频带 的分配位	模型列表	ArModel 索引	编码频带 的分配位	模型列表
0	0	表 6.1	16	8	表 6.16

1	-	未用	17	8	表 6.17
2	1	表 6.2	18	9	表 6.18
3	1	表 6.3	19	9	表 6.19
4	2	表 6.4	20	10	表 6.20
5	2	表 6.5	21	10	表 6.21
6	3	表 6.6	22	11	表 6.22
7	3	表 6.7	23	11	表 6.23
8	4	表 6.8	24	12	表 6.24
9	4	表 6.9	25	12	表 6.25
10	5	表 6.10	26	13	表 6.26
11	5	表 6.11	27	13	表 6.27
12	6	表 6.12	28	14	表 6.28
13	6	表 6.13	29	14	表 6.29
14	7	表 6.14	30	15	表 6.30
15	7	表 6.15	31	15	表 6.31

计算偏移值与所有算术编码模型系数之间的差值, 随后利用表 4.3 所列模型对差分符号进行算术编码。这里, 在表 4.3 所列的四个模型中, ArModel_model 的值表示所使用的模型, 并以 2 比特的长度将该模型存储在位流中。偏移值为存储在位流中的长度为 5 比特的 min_ArModel 的值。按照编码过程的相反次序对差分符号进行解码, 随后将差分符号累加到偏移值中, 以便恢复算术编码模型系数。

以下伪码说明对各增强层内的算术编码模型系数以及 ArModel[cband] 进行解码的方法。

```

for(ch=0; ch<nch; ch++)
    for(sfb=layer_sfb[layer]; sfb<layer_sfb[layer+1]; sfb++)
        for(g=0; g<num_window_group; g++){
            band = (sfb*num_window_group) + g
            for(i=0; swb_offset[band]; i<swb_offset[band+1]; i+=4){
                cband = index2cb(g,i);
            }
        }
    
```

```

        if(!decode_cband[ch][g][cband]){
            ArModel[g][cband] = min_ArModel + arithmetic_decoding();
            decode_cband[ch][g][cband] = 1;
        }
    }
}

```

这里，`layer_sfb[layer]`是对各增强层中的算术编码模型系数进行解码所需的起始比例因子频带，而 `layer_sfb[layer+1]`是结束比例因子频带。`decode_cband[ch][g][cband]`是一个标志，该标志表示是（1）否（0）对算术编码模型进行过解码。

2.1.6.位片式数据的解码

以位片顺序形成量化序列。根据四维向量的状态，将每个四维向量划分为两个子向量。为了进行有效压缩，以无损编码方式对两个子向量进行算术编码。确定对各编码频带进行算术编码所使用的模型。将以上信息存储到 `ArModel` 中。

如表 6.1 到表 6.31 所示，每个算术编码模型是由几个低阶模型组成的。利用某个低阶模型对子向量进行编码。根据需要编码的子向量的位数、向量的有效值或各采样的编码状态，对低阶模型进行分类。根据需要编码的向量的位位置确定向量的有效值。换句话说，随着位片信息是 MSB 的位片信息、还是下一 MSB 的位片信息抑或是 LSB 的位片信息的不同，向量的有效值不同。MSB 具有最高有效值，LSB 具有最低有效值。当从 MSB 到 LSB 进行向量编码时，更新各采样的编码状态值。首先，将编码状态值初始化为 0。然后，遇到非零比特值时，就将编码状态值改为 1。

[表 6.1]BSAC 算术模型 0

分配位 = 0

BSAC 算术模型 1

未用

[表 6.2]BSAC 算术模型 2

分配位 = 1

snf	pre_state	维	累加频率
-----	-----------	---	------

1	0	4	14858, 13706, 12545, 11545, 10434, 9479, 8475, 7619, 6457, 5456, 4497, 3601, 2600, 1720, 862, 0
---	---	---	----------------------------------------------------------------------------------------------------------

[表 6.3]BSAC 算术模型 3

分配位 = 1

snf	pre_state	维	累加频率
1	0	4	5476, 4279, 3542, 3269, 2545, 2435, 2199, 2111, 850, 739, 592, 550, 165, 21, 0

[表 6.4]BSAC 算术模型 4

分配位 = 2

snf	pre_state	维	累加频率
2	0	4	4299,3445,2583,2473,1569,1479,1371 ,1332,450,347,248,219,81,50,15, 0
1	0	4	15290,14389,13434,12485,11559,106 27,9683,8626,7691,5767,4655,3646, 2533,1415,0
		3	15139,13484,11909,9716,8068,5919, 3590,0
		2	14008,10384,6834,0
		1	11228,0
	1	4	10355,9160,7553,7004,5671,4902, 4133,3433,1908,1661,1345,1222, 796,714,233,0
		3	8328,6615,4466,3586,1759,1062, 321,0
		2	4631,2696,793,0
		1	968,0

[表 6.5]BSAC 算术模型 5

分配位 = 2

snf	pre_state	维	累加频率
2	0	4	3119,2396,1878,1619,1076,1051, 870,826,233,231,198,197, 27,26,1,0
1	0	4	3691,2897,2406,2141,1752,1668, 1497,1404,502,453,389,368, 131,102,18,0
		3	11106,8393,6517,4967,2739,2200, 608,0
		2	10771,6410,2619,0
		1	6112,0
	1	4	11484,10106,7809,7043,5053,3521, 2756,2603,2296,2143,1990,1531, 765,459,153,0
		3	10628,8930,6618,4585,2858,2129, 796,0
		2	7596,4499,1512,0
		1	4155,0

[表 6.6]BSAC 算术模型 6

分配位 = 3

snf	pre_state	维	累加频率
3	0	4	2845,2371,1684,1524,918,882,760,72 9,200,198,180,178,27,25,1,0
2	0	4	1621,1183,933,775,645,628,516,484,2 10,207,188,186,39,35,1,0
		3	8800,6734,4886,3603,1326,1204,104, 0

		2	8869,5163,1078,0
		1	3575,0
	1	4	12603,12130,10082,9767,8979,8034,7 404, 6144,4253,3780,3150,2363,1575,945, 630,0
		3	10410,8922,5694,4270,2656,1601,533 ,0
		2	8459,5107,1670,0
		1	4003,0
1	0	4	5185,4084,3423,3010,2406,2289,2169 ,2107,650,539,445,419,97,61,15,0
		3	13514,11030,8596,6466,4345,3250,12 94,0
		2	13231,8754,4635,0
		1	9876,0
	1	4	14091,12522,11247,10299,8928,7954, 6696,6024,4766,4033,3119,2508,1594 ,1008,353,0
		3	12596,10427,7608,6003,3782,2580,92 8,0
		2	10008,6213,2350,0
		1	5614,0

[表 6.7]BSAC 算术模型 7

分配位 = 3

snf	pre_state	维	累加频率
3	0	4	3833,3187,2542,2390,1676,1605,1385 ,1337,468,434,377,349,117,93,30,0
2	0	4	6621,5620,4784,4334,3563,3307,2923

			,2682,1700,1458,1213,1040,608,431,191,0
		3	11369,9466,7519,6138,3544,2441,1136,0
		2	11083,7446,3439,0
		1	8823,0
	1	4	12027,11572,9947,9687,9232,8126,7216,6176,4161,3705,3055,2210,1235,780,455,0
		3	9566,7943,4894,3847,2263,1596,562,0
		2	7212,4217,1240,0
		1	3296,0
1	0	4	14363,13143,12054,11153,10220,9388,8609,7680,6344,5408,4578,3623,2762,1932,1099,0
		3	14785,13256,11596,9277,7581,5695,3348,0
		2	14050,10293,6547,0
		1	10948,0
	1	4	13856,12350,11151,10158,8816,7913,6899,6214,4836,4062,3119,2505,1624,1020,378,0
		3	12083,9880,7293,5875,3501,2372,828,0
		2	8773,5285,1799,0
		1	4452,0

[表 6.8]BSAC 算术模型 8

分配位 = 4

snf	pre_state	维	累加频率
4	0	4	2770,2075,1635,1511,1059,1055,928,923,204,202,190,188,9,8,1,0
3	0	4	1810,1254,1151,1020,788,785,767,758,139,138,133,132,14,13,1,0
		3	7113,4895,3698,3193,1096,967,97,0
		2	6858,4547,631,0
		1	4028,0
	1	4	13263,10922,10142,9752,8582,7801,5851,5071,3510,3120,2730,2340,1560,780,390,0
		3	12675,11275,7946,6356,4086,2875,1097,0
		2	9473,5781,1840,0
		1	3597,0
2	0	4	2600,1762,1459,1292,989,983,921,916,238,233,205,202,32,30,3,0
		3	10797,8840,6149,5050,2371,1697,483,0
		2	10571,6942,2445,0
		1	7864,0
	1	4	14866,12983,11297,10398,9386,8683,7559,6969,5451,4721,3484,3007,1882,1208,590,0
		3	12611,10374,8025,6167,4012,2608,967,0
		2	10043,6306,2373,0
		1	5766,0
1	0	4	6155,5057,4328,3845,3164,2977,2728

			,2590,1341,1095,885,764,303,188,64,0
		3	12802,10407,8142,6263,3928,3013,1225,0
		2	13131,9420,4928,0
		1	10395,0
	1	4	14536,13348,11819,11016,9340,8399,7135,6521,5114,4559,3521,2968,1768,1177,433,0
		3	12735,10606,7861,6011,3896,2637,917,0
		2	9831,5972,2251,0
		1	4944,0

[表 6.9]BSAC 算术模型 9

分配位 = 4

snf	pre_state	维	累加频率
4	0	4	3383,2550,1967,1794,1301,1249,1156,1118,340,298,247,213,81,54,15,0
3	0	4	7348,6275,5299,4935,3771,3605,2962,2818,1295,1143,980,860,310,230,75,0
		3	9531,7809,5972,4892,2774,1782,823,0
		2	11455,7068,3383,0
		1	9437,0
	1	4	12503,9701,8838,8407,6898,6036,4527,3664,2802,2586,2371,2155,1293,431,215,0
		3	11268,9422,6508,5277,3076,2460,145

			7,0
		2	7631,3565,1506,0
		1	2639,0
2	0	4	11210,9646,8429,7389,6252,5746,514 0,4692,3350,2880,2416,2014,1240,85 1,404,0
		3	12143,10250,7784,6445,3954,2528,12 28,0
		2	10891,7210,3874,0
		1	9537,0
	1	4	14988,13408,11860,10854,9631,8992, 7834,7196,5616,4793,3571,2975,1926 ,1212,627,0
		3	12485,10041,7461,5732,3669,2361,94 0,0
		2	9342,5547,1963,0
		1	5410,0
1	0	4	14152,13258,12486,11635,11040,102 90,9740,8573,7546,6643,5903,4928,4 005,2972,1751,0
		3	14895,13534,12007,9787,8063,5761,3 570,0
		2	14088,10108,6749,0
		1	11041,0
	1	4	14817,13545,12244,11281,10012,895 2,7959,7136,5791,4920,3997,3126,21 05,1282,623,0
		3	12873,10678,8257,6573,4186,2775,10 53,0

		2	9969,6059,2363,0
		1	5694,0

[表 6.10]BSAC 算术模型 10

分配位(Abit) = 5

snf	pre_state	维	累加频率
Abit	0	4	2335,1613,1371,1277,901,892,941,833,141,140,130,129,24,23,1,0
Abit-1	0	4	1746,1251,1038,998,615,611,583,582,106,104,101,99,3,2,1,0
		3	7110,5230,4228,3552,686,622,46,0
		2	6101,2575,265,0
		1	1489,0
	1	4	13010,12047,11565,11803,9637,8673,6264,5782,4336,3855,3373,2891,2409,1927,963,0
		3	10838,10132,8318,7158,5595,3428,2318,0
		2	8209,5197,1287,0
		1	4954,0
Abit-2	0	4	2137,1660,1471,1312,1007,1000,957,951,303,278,249,247,48,47,1,0
		3	9327,7413,5073,4391,2037,1695,205,0
		2	8658,5404,1628,0
		1	5660,0
	1	4	13360,12288,10727,9752,8484,7899,7119,6631,5363,3900,3023,2535,1852,1267,585,0
		3	13742,11685,8977,7230,5015,3427,1132,0

		2	10402,6691,2828,0
		1	5298,0
Abit-3	0	4	4124,3181,2702,2519,1949,1922,1733, 1712,524,475,407,78,52,15,0
		3	10829,8581,6285,4865,2539,1920,594, 0
		2	11074,7282,3092,0
		1	8045,0
	1	4	14541,13343,11637,10862,9328,8783, 7213, 6517,5485,5033,4115,3506,2143,1555, 509, 0
		3	13010,11143,8682,7202,4537,3297,12 21,0
		2	9941,5861,2191,0
		1	5340,0
其 他 snf	0	4	9845,8235,7126,6401,5551,5131,4664, 4320,2908,2399,1879,1506,935,603,27 7,0
		3	13070,11424,9094,7203,4771,3479,14 86,0
		2	13169,9298,5406,0
		1	10371,0
	1	4	14766,13685,12358,11442,10035,9078 ,7967,7048,5842,5006,4058,3400,2350 ,1612,659,0
		3	13391,11189,8904,7172,4966,3183,13 83,0

		2	10280,6372,2633,0
		1	5419,0

[表 6.11]BSAC 算术模型 11

分配位(Abit) = 5

snf	pre_state	维	累加频率
Abit	0	4	2872,2294,1740,1593,1241,1155,1035, 960,339,300,261,247,105,72,34,0
Abit-1	0	4	3854,3090,2469,2276,1801,1685,1568, 1505,627,539,445,400,193,141,51,0
		3	10654,8555,6875,4976,3286,2229,826, 0
		2	10569,6180,2695,0
		1	6971,0
	1	4	11419,11170,10922,10426,7943,6950,3 723,3475,1737,1489,1241,992,744,496 ,248,0
		3	11013,9245,6730,4962,3263,3263,1699 ,883,0
		2	6969,4370,1366,0
		1	3166,0
Abit-2	0	4	9505,8070,6943,6474,5305,5009,4290, 4029,2323,1911,1591,1363,653,443,21 7,0
		3	11639,9520,7523,6260,4012,2653,1021 ,0
		2	12453,8284,4722,0
		1	9182,0
	1	4	13472,12294,10499,9167,7990,7464,65 65,6008,4616,3747,2818,2477,1641,10

			84,557,0
		3	13099,10826,8476,6915,4488,2966,1223,0
		2	9212,5772,2053,0
		1	4244,0
Abit-3	0	4	14182,12785,11663,10680,9601,8748,8135,7353,6014,5227,4433,3727,2703,1818,866,0
		3	13654,11814,9714,7856,5717,3916,2112,0
		2	12497,8501,4969,0
		1	10296,0
	1	4	15068,13770,12294,11213,10230,9266,8439,7438,6295,5368,4361,3620,2594,1797,895,0
		3	13120,10879,8445,6665,4356,2794,1047,0
		2	9311,5578,1793,0
		1	4695,0
其 他 snf	0	4	15173,14794,14359,13659,13224,12600,11994,11067,10197,9573,9081,7624,6697,4691,3216,0
		3	15328,13985,12748,10084,8587,6459,4111,0
		2	14661,11179,7924,0
		1	11399,0
	1	4	14873,13768,12458,11491,10229,9164,7999,7186,5992,5012,4119,3369,2228,1427,684,0

		3	13063,10913,8477,6752,4529,3047,124 1,0
		2	10101,6369,2615,0
		1	5359,0

[表 6.12]BSAC 算术模型 12

与 BSAC 算术模型 10 相同，但分配位= 6

[表 6.13]BSAC 算术模型 13

与 BSAC 算术模型 11 相同，但分配位= 6

[表 6.14]BSAC 算术模型 14

与 BSAC 算术模型 10 相同，但分配位= 7

[表 6.15]BSAC 算术模型 15

与 BSAC 算术模型 11 相同，但分配位= 7

[表 6.16]BSAC 算术模型 16

与 BSAC 算术模型 10 相同，但分配位= 8

[表 6.17]BSAC 算术模型 17

与 BSAC 算术模型 11 相同，但分配位= 8

[表 6.18]BSAC 算术模型 18

与 BSAC 算术模型 10 相同，但分配位= 9

[表 6.19]BSAC 算术模型 19

与 BSAC 算术模型 11 相同，但分配位= 9

[表 6.20]BSAC 算术模型 20

与 BSAC 算术模型 10 相同，但分配位= 10

[表 6.21]BSAC 算术模型 21

与 BSAC 算术模型 11 相同，但分配位=10

[表 6.22]BSAC 算术模型 22

与 BSAC 算术模型 10 相同，但分配位= 11

[表 6.23]BSAC 算术模型 23

与 BSAC 算术模型 11 相同，但分配位= 11

[表 6.24]BSAC 算术模型 24

与 BSAC 算术模型 10 相同，但分配位= 12

[表 6.25]BSAC 算术模型 25

与 BSAC 算术模型 11 相同，但分配位= 12

[表 6.26]BSAC 算术模型 26

与 BSAC 算术模型 10 相同，但分配位= 13

[表 6.27]BSAC 算术模型 27

与 BSAC 算术模型 11 相同，但分配位= 13

[表 6.28]BSAC 算术模型 28

与 BSAC 算术模型 10 相同，但分配位= 14

[表 6.29]BSAC 算术模型 29

与 BSAC 算术模型 11 相同，但分配位= 14

[表 6.30]BSAC 算术模型 30

与 BSAC 算术模型 10 相同，但分配位= 15

[表 6.31]BSAC 算术模型 31

与 BSAC 算术模型 11 相同，但分配位=15

两个子向量为 1 到 4 维向量。按照从 MSB 到 LSB，从较低频率分量到较高频率分量的次序，对以上子向量进行算术编码。在将位片式数据传输到编码频带组中各编码频带之前，按照从低频到高频的次序，在位流内预先存储算术编码中所使用的算术编码模型系数。

对各位片式数据进行算术编码以得到编码字系数。通过利用以下伪码进行位耦合，将以上系数恢复为原始量化数据。

‘pre_state[]’为一状态，该状态表示当前解码值是否为 0。‘snf’为经过解码的向量的有效值。‘idx0’为编码字系数，其预状态为 0。

‘idx1’为编码字系数，其预状态为 1。‘dec_sample[]’为经过解码的数据。‘start_i’为经过解码的向量的起始频率行号。

```
for(i=start_i; i<(start_i + 4); i++){
    if(pre_state[i]){
        if(idx1 & 0x01)
            dec_sample[i] |= (1<<(snf-1))
            idx1>>=1;
    }
    else{
        if(idx0 & 0x01)
            dec_sample[i] |= (1<<(snf-1))
            idx0>>=1;
    }
}
```

在按照从 MSB 到 LSB 的次序，对量化频率分量的位片式数据进行编码时，对非零频率系数的符号位进行算术编码。用 1 表示负 (-) 符号位，用 0 表示正 (+) 符号位。

因此，如果在解码器内对位片式数据进行算术解码，并且首先遇到经

过算术解码的非零位值，则随后为位流中的符号信息，即 `acode_sign`。借助表 5.9 所列模型，利用以上信息对 `sign_bit` 进行算术解码。如果 `sign_bit` 为 1，则按如下方式，通过耦合分离数据将符号信息赋予量化数据 (`y`)。

```
if(y != 0)
    if(sign_bit == 1)
        y = -y
```

2.2.M/S 立体声处理部件（可选模块）

根据位流中包含的标志以及 `ms_used[]`，确定各比例因子频带是否采用了 M/S 立体声处理部件。如果采用了该部件，就采用 ACC 中说明的相同过程进行 M/S 立体声处理。

2.3.预测部件（可选模块）

根据位流中包含的标志以及 `prediction_present`，确定各比例因子频带是否采用了预测部件。如果采用了该部件，就采用 ACC 中说明的相同过程进行预测。

2.4.立体声强度处理部件（可选模块）

根据位流中包含的标志以及 `stereo_info`，确定各比例因子频带是否采用了立体声强度处理部件。如果采用了该部件，就采用 ACC 中说明的相同过程进行立体声强度处理。

2.5.TNS 部件（可选模块）

根据位流中包含的标志以及 `tns_present`，确定是否采用了 TNS 部件。如果采用了该部件，就采用 ACC 中说明的相同过程进行 TNS 处理。

2.6.反向量化

反向量化部件将经过解码的比例因子与量化数据恢复为具有原始数值的信号。ACC 标准说明了反向量化过程。

2.7.频率/时间映射

频率/时间映射部件将频域音频信号反演为时域信号，以使用户重播。ACC 标准定义了将频域信号映射为时域信号的公式，此外，ACC 标准还说明了各种术语，如与映射各的窗口。

本发明提供与常规编码器类似的性能，从而能够处理单声道信号和立体声信号以满足不同用户需求，同时形成可伸缩的位流，其中常规编码器

仅仅考虑在较高的位率进行压缩。换句话说，根据用户请求，将各层的位率信息融合到某位流中而无需叠加，从而提供具有较好音质的位流。此外，在发送端和接收端之间不需要转换器。另外，本发明可以适应任何传输通道状态以及各种用户请求。

此外，可伸缩性适用于立体声信号，也适用于单声道信号。

可以将本发明沿用于具有以下模块的常规音频编码/解码装置，从而改进在不同位率下的性能，其中该模块用于提高编码/解码的效率。

此外，在本发明中，在使用 ACC 标准中所采用的基本模块（如时间/频率映射模块或量化模块）时，仅利用位片式编码方法取代无损编码模块，以便提供可伸缩性。

由于位流是可伸缩的，所以某位流可以包含具有各种位率的不同位流。与常规编码器不同，根据本发明的可伸缩的编码器具有精细分层的增强层，从而推广了应用范围。

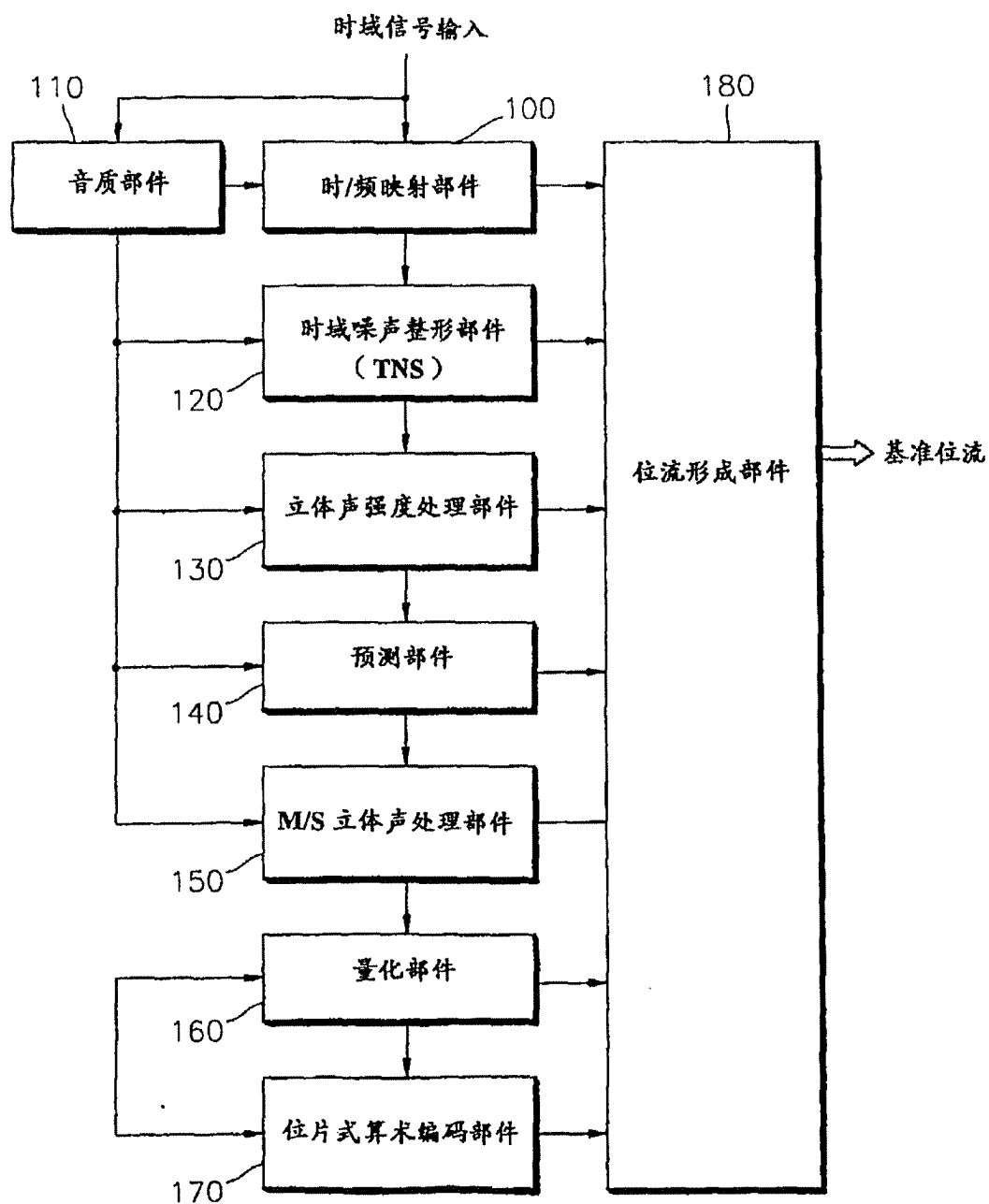
此外，与其他可伸缩的编码解码器不同，在较高位率提供优良音质。

如果联合本发明与 ACC 标准，则在顶层位率下，也能够获得几乎相同的音质。

根据本发明，在利用诸如 MPEG-2 ACC 标准之类的常规音频算法时，只是无损编码部件与常规部件不同。因此，以 ACC 位流方式，对频域上的量化信号进行解码，并且能够根据经过解码的信号生成可伸缩的 BSAC 位流。换句话说，提供无损解码。此外，可以按照相反次序，根据可伸缩的 BSAC 位流生成 ACC 位流。由于以上功能，所以能够根据其环境转换使用仅仅为提高编码效率而生成的各种 ACC 位流。因此，为了提供可伸缩性，单个编码装置并不需要两倍或三倍的工作量来生成提供可伸缩性所需的位流。

此外，本发明具有良好的编码效率，即，如同常规编码技术一样，在固定位率呈现最佳性能，并且本发明涉及编码/解码方法和装置，其中恢复为适应多媒体技术的来临而编码的位率。此外，根据本发明，可以在某个位流内表示所有增强层的位率数据。因此，根据用户解码器的性能以及传输通道的带宽/拥塞或者根据用户请求，可以控制位率的高低以及其复杂性。

图 1



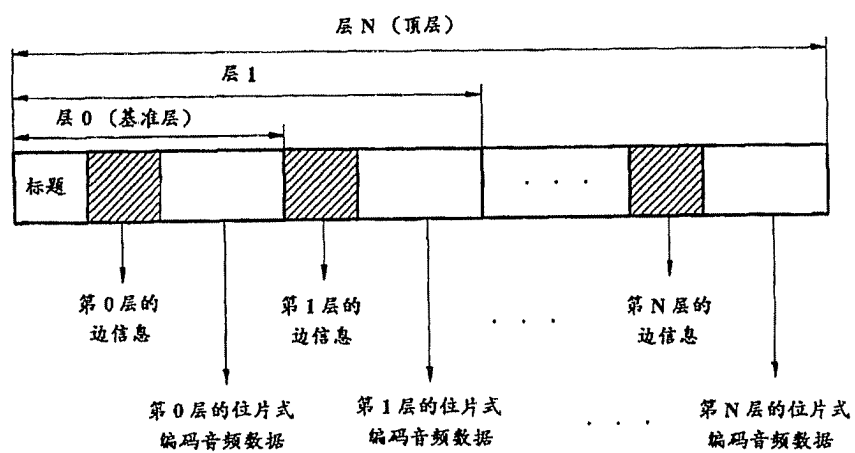
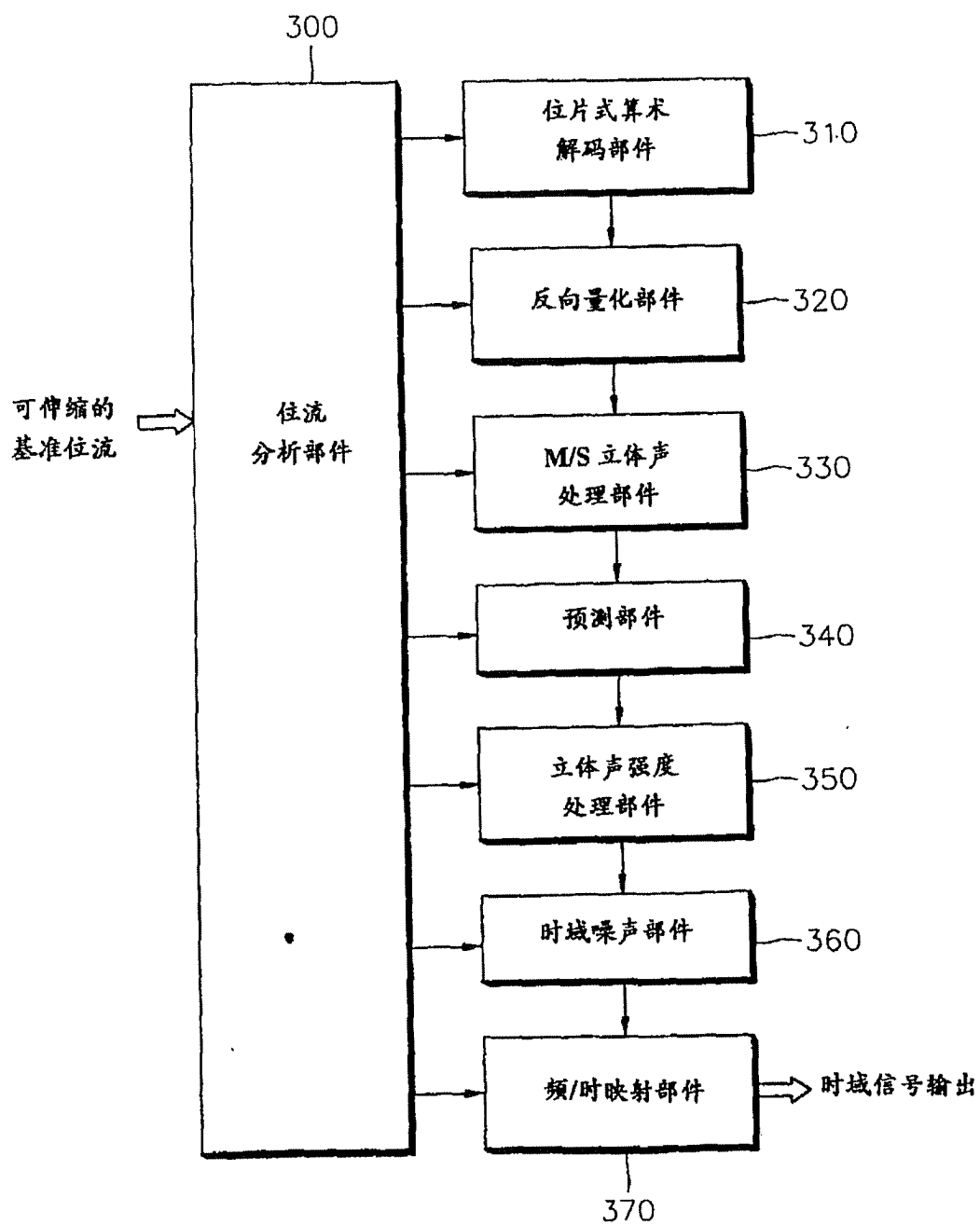


图 2

图 3



频谱系数 →

sfb0	sfb0	sfb0	...	sfb(num_sfb-1)
------	------	------	-----	----------------

图 4

频谱系数 →

B0 0..3	B1 0..3	B2 0..3	...	B7 0..3	B0 4..7	B1 4..7	B2 4..7	...	B7 4..7	B1 80...83	B2 80...83	B3 80...83	...	B7 80...83	...
------------	------------	------------	-----	------------	------------	------------	------------	-----	------------	---------------	---------------	---------------	-----	---------------	-----

图 5